WHAT IS CLAIMED IS:

1   1. A method of testing a product configuration in a system for generating

2   product configurations, the system including at least one rule defining a relationship

3   between at least two parts, the product configuration including a plurality of parts, the

4   method comprising:

5   entering a test case, wherein the test case selects at least one part to include in

6   the product configuration; and

7   processing the at least one rule to determine whether the at least one part

8   selected in the test case conflicts with the plurality of parts previously

9   included in the product configuration.

1   2. The method, as set forth in claim 1, wherein processing the at least one rule

2   to determine whether the at least one part selected in the test case conflicts with the

3   plurality of parts previously included in the product configuration, further includes:

4   initializing the system with a part state;

5   inputting the at least one part selection; and

6   listening to state change events in the system to detect when a state change

7   event occurs that results in the system being in the initialized part state.

1   3. The method, as set forth in claim 2, wherein processing the at least one rule

2   to determine whether the at least one part selected in the test case conflicts with the

3   plurality of parts previously included in the product configuration, further includes:

4   generating a cause that explains the part state in terms of the state change

5   event.

1   4. The method, as set forth in claim 3, wherein processing the at least one rule

2   to determine whether the at least one part selected in the test case conflicts with the

3   plurality of parts previously included in the product configuration, further includes:

4   generating a new part state for each part associated with the cause.

1   5. The method, as set forth in claim 4, wherein processing the at least one rule

2   to determine whether the at least one part selected in the test case conflicts with the

3   plurality of parts previously included in the product configuration, further includes:

4   determining the causes that explain the new part states in terms of the state

5   change event.

1   6. The method, as set forth in claim 5, further comprising:

2   generating a cause tree wherein the root of the cause tree is the initial part

3   state, and leaves of the tree are the user's selections of parts.

1   7. The method, as set forth in claim 6, further comprising:

2   generating an explanation of the part state wherein the part selections are the

3   root of the explanation and the causes follow from the part selections.

1   8. The method, as set forth in claim 7, wherein the explanation is based on

2   selection of a part.

1   9. The method, as set forth in claim 7, wherein the explanation is based on

2   execution of a rule.

1   10. The method, as set forth in claim 7, wherein the explanation is based on a

2   part being in two states at the same time.

1   11. The method, as set forth in claim 7, wherein the explanation is based on a

2   requires choice rule that cannot be satisfied.

1   12. The method, as set forth in claim 7, wherein the explanation is based on a

2   look ahead process.

1   13. The method, as set forth in claim 7, further comprising:

2   sorting the tree by iteration number, wherein the iteration number of a part

3   state is determined by measuring the longest distance between the part

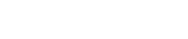4   state and the cause corresponding to the part state.

14. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for testing a product configuration in a system for generating product configurations, the system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the computer readable program code including:

computer readable program code configured to cause a computer to allow a user to enter a test case, wherein the test case selects at least one part to include in the product configuration; and

computer readable program code configured to cause a computer to process the at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration.

15. The article of manufacture, as set forth in claim 14, further including:

computer readable program code configured to cause a computer to initialize the system with a part state;

computer readable program code configured to cause a computer to input the at least one part selection; and

computer readable program code configured to cause a computer to listen to state change events in the system to detect when a state change event occurs that results in the system being in the initialized part state.

16. The article of manufacture, as set forth in claim 15, further including:

computer readable program code configured to cause a computer to generate a cause that explains the part state in terms of the state change event.

17. The article of manufacture, as set forth in claim 16, further including:

computer readable program code configured to cause a computer to generate a new part state for each part associated with the cause.

18. The article of manufacture, as set forth in claim 17, further including:

2   computer readable program code configured to cause a computer to determine

3   the causes that explain the new part states in terms of the state change

4   event.

1   19. The article of manufacture, as set forth in claim 18, further comprising:

2   computer readable program code configured to cause a computer to generate a

3   cause tree wherein the root of the cause tree is the initial part state, and

4   leaves of the tree are the user's selections of parts.

1   20. The article of manufacture, as set forth in claim 19, further comprising:

2   computer readable program code configured to cause a computer to generate

3   an explanation of the part state wherein the part selections are the root

4   of the explanation and the causes follow from the part selections.

1   21. The article of manufacture, as set forth in claim 20, wherein the

2   explanation is based on selection of a part.

1   22. The article of manufacture, as set forth in claim 20, wherein the

2   explanation is based on execution of a rule.

1   23. The article of manufacture, as set forth in claim 20, wherein the

2   explanation is based on a part being in two states at the same time.

1   24. The article of manufacture, as set forth in claim 20, wherein the

2   explanation is based on a requires a choice rule that cannot be satisfied.

1   25. The article of manufacture, as set forth in claim 20, wherein the

2   explanation is based on a look ahead process.

1   26. The article of manufacture, as set forth in claim 20, further comprising:

2   computer readable program code configured to cause a computer to sort the

3   tree by iteration number, wherein the iteration number of a part state is

4   determined by measuring the longest distance between the part state

5   and the cause corresponding to the part state.

1   27. An apparatus for testing a product configuration generated by a product

2 configuration system, comprising:

3   at least one rule defining a relationship between at least two parts in the

4   product configuration;

5   a test case pertaining to at least one part to include in the product

6   configuration; and

7   a processor coupled to receive the at least one rule and the test case, wherein

8   the processor is operable to determine whether the at least one part in

9   the test case conflicts with the plurality of parts previously included in

10   the product configuration according to the at least one rule.

1   28. The apparatus, as set forth in claim 27, wherein the processor is further

2 operable to:

3   initialize the configuration system with a part state;

4   to input the at least one part selection;

5   to listen to state change events in the system; and

6   to detect when a state change event occurs that results in the configuration

7   system being in the initialized part state.

1   29. The apparatus, as set forth in claim 28, wherein the processor is further

2 operable to:

3   generate a cause that explains the part state in terms of the state change event.

1   30. The apparatus, as set forth in claim 29, wherein the processor is further

2 operable to:

3   generate a new part state for each part associated with the cause.

1   31. The apparatus, as set forth in claim 30, wherein the processor is further

2 operable to:

3   generate a cause tree wherein the root of the cause tree is the initial part state,

4   and leaves of the tree are the user's selections of parts.
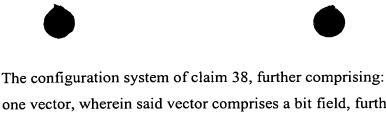
1   32. The apparatus, as set forth in claim 30, wherein the processor is further

2   operable to:

3   generate an explanation of the part state wherein the part selections are the

4   root of the explanation and the causes follow from the part selections.

1   33. The apparatus, as set forth in claim 32, wherein the explanation is based

2   on execution of a rule.

1   34. The apparatus, as set forth in claim 32, wherein the explanation is based

2   on a part being in two states at the same time.

1   35. The apparatus, as set forth in claim 32, wherein the explanation is based

2   on a requires a choice rule that cannot be satisfied.

1   36. The apparatus, as set forth in claim 32, wherein the explanation is based

2   on a look ahead process.

1   37. The apparatus, as set forth in claim 30, wherein the processor is further

2   operable to:

3   sort the tree by iteration number, wherein the iteration number of a part state is

4   determined by measuring the longest distance between the part state

5   and the cause corresponding to the part state.

1   38.   A configuration system comprising:

2   a modified database, wherein the modified database is based on the results of

3   testing a product selection using a test case, wherein the test case

4   includes at least one part selection, and the expected state of the at least

5   one selected part.

1   39.   The configuration system of claim 38, wherein the test case further

2   includes the product selection.

1   40.   The configuration system of claim 38, further comprising:

2   at least one vector, wherein said vector comprises a bit field, further wherein

3   the bit field comprises bits that represent elements in a configuration.

1   41.   The configuration system of claim 40, wherein the number of bits in

2   the bit field is equal to the total number of elements and an element's bit can be set or

3   reset to specify the state of the element in the configuration.

1   42.   The configuration system of claim 40, wherein the vector specifies

2   whether an element has been selected by the user during the configuration.

1   43.   The configuration system of claim 40, wherein excluded vectors

2   identify whether an element is excluded from a configuration.

1   44.   The configuration system of claim 40, wherein removed vectors

2   identify whether an element is removed from a configuration.

1   45.   The configuration system of claim 40, wherein the vector identifies

2   whether an element is selectable.

1   46.   A database comprising:

2   at least one table, wherein said table represents relationships between elements

3   in a configuration; and

4   at lease one modified rule, wherein the rule is modified based on the results of

5   testing a product selection.

1   47.   The database of claim 46, wherein said table represents "includes"

2   relationships between elements in a configuration.

1   48.   The database of claim 46, wherein said table represents "excludes"

2   relationships between elements in a configuration.

1 49.   The database of claim 46, wherein said table represents "removes"

2 relationships between elements in a configuration.

1 50.   The database of claim 46, wherein said table represents "requires

2 choice" relationships between elements in a configuration.

1 51.   The database of claim 50, wherein the representation of "requires

2 choice" relationships includes a pointer to a group table that includes a bit vector that

3 identifies the elements that are contained in the group from which a choice is to be

4 made.

1 52.   The database of claim 50, wherein the representation of "requires

2 choice" relationships includes minimum and maximum designations to identify the

3 minimum and maximum number of group members that are to be selected to satisfy

4 the "requires choice" relationship.

1 53.   The database of claim 46, wherein said table includes a left-hand side

2 and a right-hand side.

1 54.   The database of claim 53, wherein the left-hand side includes a bit

2 vector that contains bits corresponding to elements.

1 55.   The database of claim 53, wherein the right-hand side includes one or

2 more bit vectors that represent configuration elements.

1 56. A test case for testing a product configuration generated by a product

2 configuration system, comprising:

3   a product selection;

4   at least one part selection; and

5   an expected state of the selected part based on one or more rules.

1 57. A method for identifying an invalid configuration generated by a product

2 configuration system, comprising:

29

3   selecting a product;

4   selecting at least one part; and

5   generating a part state of the selected part based on one or more rules.

1   58. The method as set forth in claim 57, further comprising:

2   generating a cause that explains the part state in terms of a state change event.

1   59. The method as set forth in claim 57, further comprising:

2   detecting an error in the part state.

1   60. The method as set forth in claim 57, further comprising:

2   detecting when the at least one part is put in more than one state at the same

3   time.

1   61. The method as set forth in claim 57, further comprising:

2   determining whether the product is selectable.

1   62. The method as set forth in claim 57, further comprising:

2   detecting when the at least one part is put in more than one state at the same

3   time.

1   63. The method as set forth in claim 57, further comprising:

2   reporting the state of the product as not selectable when selection of the

3   product would conflict with the rule.

1   64. The method as set forth in claim 57, further comprising:

2   determining sets of parts that are excluded or deleted based on the product.

1   65. The method as set forth in claim 57, further comprising:

2   detecting when a state change event occurs that results in the system being in

3   the initialized part state.

1   66. The method as set forth in claim 65, further comprising:

2   generating a cause that explains the part state in terms of the state change

3   event.

1   67. The method, as set forth in claim 66, further comprising:

2   generating a new part state for each part associated with the cause.

1   68. The method, as set forth in claim 67, further comprising:

2   generating a cause tree wherein the root of the cause tree is the initial part

3   state, and leaves of the tree are the user's selections of parts.

1   69. The method, as set forth in claim 68, further comprising:

2   generating an explanation of the part state wherein the part selections are the

3   root of the explanation and the causes follow from the part selections.

1   70. An apparatus for testing a product configuration generated by a product

2   configuration system, comprising:

3   means for defining a relationship between at least two parts in the product

4   configuration;

5   means for defining a test case for at least one part to include in the product

6   configuration; and

7   means for determining whether the at least one part in the test case conflicts

8   with the plurality of parts previously included in the product

9   configuration according to at least one rule.

1   71. The apparatus, as set forth in claim 70, further comprising:

2   means for initializing the configuration system with a part state;

3   means for detecting a state change event in the system; and

4   means for detecting when a state change event occurs that results in the

5   configuration system being in the initialized part state.

1   72. The apparatus, as set forth in claim 71, further comprising:

2   means for generating a cause that explains the part state in terms of the state

3   change event.

1   73. The apparatus, as set forth in claim 72, further comprising:

2   means for generating a new part state for each part associated with the cause.

1   74. The apparatus, as set forth in claim 73, further comprising:

2   means for generating a cause tree, wherein the root of the cause tree is the

3   initial part state, and leaves of the tree are the user's selections of parts.

1   75. The apparatus, as set forth in claim 73, further comprising:

2   means for generating an explanation of the part state, wherein the part

3   selections are the root of the explanation and the causes follow from

4   the part selections.

1   76. The apparatus, as set forth in claim 70, further comprising:

2   means for modifying the at least one rule when the test case conflicts with the

3   plurality of parts previously included in the product configuration.

4

5